Automatic Search of Truncated Impossible Differentials for Word-Oriented Block Ciphers

Authors: Shengbao Wu and Mingsheng Wang

Chinese Academy of Sciences, Beijing, China
wushengbao@is.iscas.ac.cn
mingsheng_wang@yahoo.com.cn

Speaker: Subhabrata Samajder

Indian Statistical Institute, Kolkata

Indocrypt 2012

Outline

- Introduction
- New Tool for Finding Impossible Differentials
- Conclusions

Outline

- **Introduction**
  - Impossible differential cryptanalysis (IDC)
  - Previous tools for finding impossible differentials (IDs) and their disadvantages
  - Our contributions
- **New Tool for Finding Impossible Differentials**
- **Conclusions**

Introduction

——Impossible differential cryptanalysis

- **Impossible Differential Cryptanalysis (IDC)**
  - One of the most popular cryptanalytic tools for block ciphers
  - Proposed by Knudsen (1998) to attack DEAL and extended by Biham et al. to analyze IDEA and Skipjack
  - Exploits differentials with probability zero to recover keys
- **General Steps of IDC**
  - Find impossible differentials
  - Guess subkey material, use a list of plaintext-ciphertext pairs to filter out all wrong keys

Introduction

——Impossible differential cryptanalysis

❑ Perform a successful IDC

- ➢ Influence factors
  - ✓ Length of impossible differentials
  - ✓ Specific input/output difference patterns
  - ✓ Strength of one-round encryption/decryption
- ➢ Find longer impossible differentials
  - ✓ The longer the impossible differential is, the better the attack will be.
- ➢ Find more impossible differentials
  - ✓ More impossible differentials we find, more probabilities we can perform a successful attack or improve known attacks.

Introduction

——Previous tools and their disadvantages

$\mathcal{U}$-method and UID-method

- In Indocrypt 2003, Kim et al. proposed the $\mathcal{U}$-method to find impossible differentials for block cipher structures with bijective round functions.
- Extended by Luo et al. (2009), and named as the UID-method
- Based on the miss-in-the-middle approach

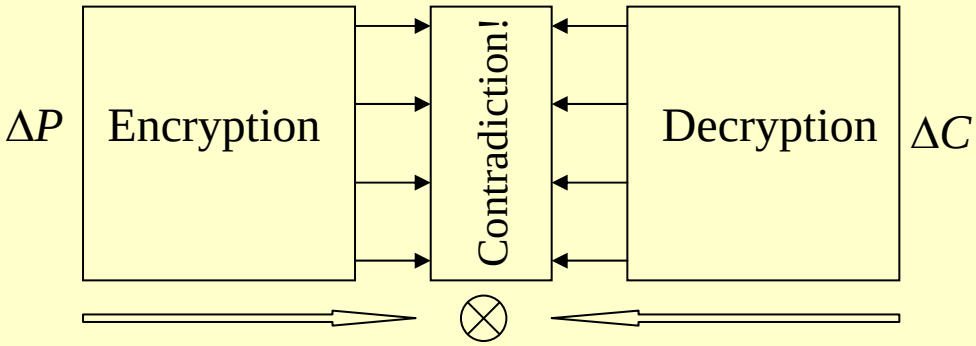


Fig 1. Basic idea of the miss-in-the-middle approach

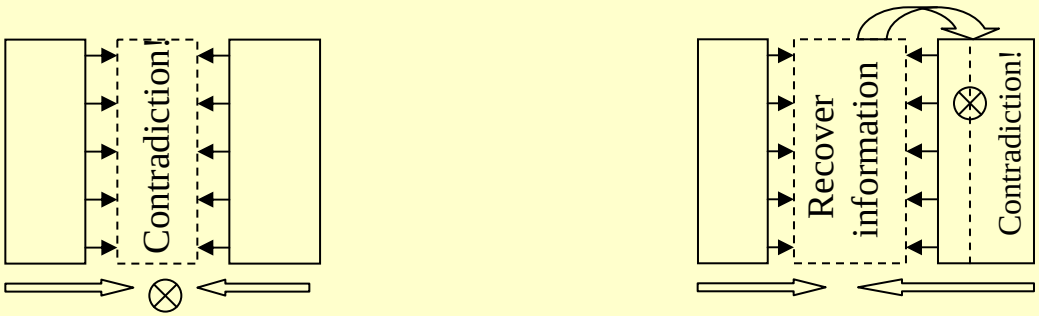
- Employed by the designers of LBlock (ACNS'11), Piccolo (CHES'11) and TWINE (SAC'12)

Introduction

——**Previous tools and their disadvantages**

❑ Disadvantages of previous tools

- ➢ Miss-in-the-middle approach limits their power



1-(a) 1-(b)

Fig 2. Miss-in-the-middle approach (1-(a)) and IDs with information feedback (1-(b))

- ➢ Failed to find the longest known IDs of many block ciphers, such as Camellia, MIBS and E2
- ➢ There is a gap between previous tools and ad hoc approaches

Introduction

——Our contributions

Our contributions

- Proposed a new tool to search IDs for word-oriented block ciphers with bijective Sboxes
- The $\mathcal{U}$-method and the UID-method are specific cases of our tool, and our tool is more powerful than them.
- Although our tool does not improve the lengths of IDs for existing block ciphers, it helps in reducing the gap between previous tools and ad hoc approaches
- Not only rediscovers the longest truncated IDs of many word-oriented block ciphers known so far, but also finds new results

Introduction

——Our contributions

Table 1. Summary of new truncated impossible differentials (ID) obtained by our tool.

Block cipher	Word unit	Previous results			In this paper		
		Round	No. of IDs	Method	Round	No. of IDs	**New IDs**
CLEFIA	Byte	9	72	ad hoc	9	72	**0**
AES	Byte	4	269,554	ad hoc	4	3,608,100	**3,338,546**
ARIA	Byte	4	156	ad hoc	4	94,416	**94,260**
Camellia*	Byte	8	3	ad hoc	8	4	**1**
E2	Byte	6	1	ad hoc	6	56	**55**
MIBS	Nibble	8	2	ad hoc	8	8	**6**
LBlock	Nibble	14	64	$\mathcal{U}$-method	14	80	**16**
Piccolo	Nibble	7	1	$\mathcal{U}$-method	7	450	**449**

Outline

- Introduction
- **New Tool for Finding Impossible Differentials**
  - Sketch of our tool
  - Difference propagation system
  - Predict information and detect contradictions
  - Algorithm complexity and experimental results
  - Discussions
- Conclusions

New Tool for Finding Impossible Differentials

——Sketch of our tool

Basic idea of our tool

- Consider an r-round block cipher, for given $(\Delta P, \Delta C)$
  - $\Delta P \rightarrow \Delta C$ is possible, if there is a differential trail started from ΔP, and ended at ΔC.
  - Otherwise, impossible
- If we view a block cipher as a system of equations, then for given $(\Delta P, \Delta C)$
  - It is an impossible differential if the corresponding system doesn't have any solution.

New Tool for Finding Impossible Differentials

——Sketch of our tool

Sketch of our tool

- Treat a block cipher as an entirety, and describe the propagation of differences as a system of equations (Difference propagation system)
- For given plaintext difference and ciphertext difference, predict information using difference propagation system iteratively, with probability one in each step
- The process of predicting information will be stopped if we find a contradiction or we can not get new information any longer.
- Note: a contradiction is found implies that the difference propagation system doesn’t have any solution under given plaintext and ciphertext difference, which also means we obtain an impossible differential.

New Tool for Finding Impossible Differentials

——Sketch of our tool

1. Build the difference propagation system of a block cipher;
2. **for** *each pair of* $(\Delta P, \Delta C)$ *we choose* **do**;
3. *index*:=true;
4. **while** *index* **do**;
5. Predict information from the difference propagation system;
6. **if** *a contradiction is found* **then**;
7. *index*:=false; **return** true;
8. **elseif** *cannot get any new information* **then**;
9. *index*:=false;
10. **end if**;
11. **end while**;
12. **end for**;

Algorithm 1. Sketch of our tool

New Tool for Finding Impossible Differentials

——Sketch of our tool

1. Build the difference propagation system of a block cipher;
2. **for** *each pair of* $(\Delta P, \Delta C)$ *we choose* **do**;
3. *index*:=true;
4. **while** *index* **do**;
5. Predict information from the difference propagation system;
6. **if** *a contradiction is found* **then**;
7. *index*:=false; **return** true;
8. **elseif** *cannot get any new information* **then**;
9. *index*:=false;
10. **end if**;
11. **end while**;
12. **end for**;

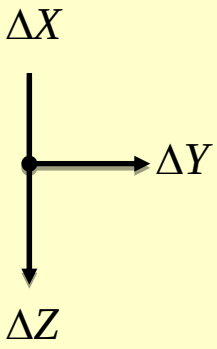
Algorithm 1. Sketch of our tool

New Tool for Finding Impossible Differentials

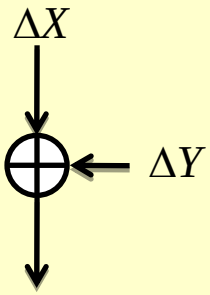
——Difference propagation system

Build equations for basic primitives

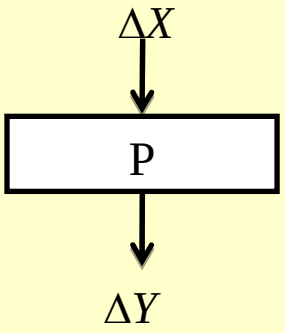
- We consider the XOR difference and suppose subkeys are XORed to the state, we can omit the key addition layers.
- Four basic primitives are widely used in word-oriented block ciphers



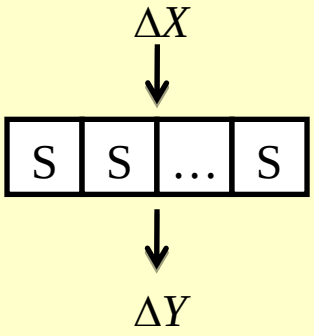
Branching



XOR



Linear permutation layer

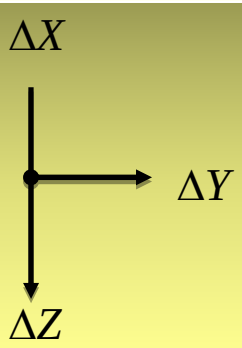
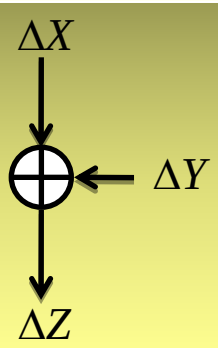
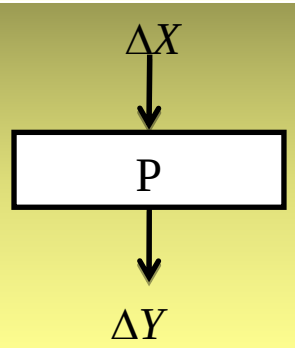
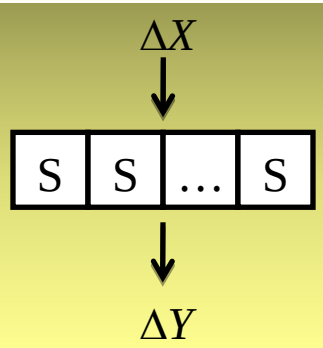


Sbox layer

New Tool for Finding Impossible Differentials

——Difference propagation system

- **Suppose** $\Delta X = (\Delta x_1, \Delta x_2, \ldots, \Delta x_n), \Delta Y = (\Delta y_1, \Delta y_2, \ldots, \Delta y_n)$ **and** $\Delta Z = (\Delta z_1, \Delta z_2, \ldots, \Delta z_n)$

Basic primitives				
Equations	$\Delta x_i \oplus \Delta y_i = 0,$ $\Delta x_i \oplus \Delta z_i = 0$	$\Delta x_i \oplus \Delta y_i \oplus \Delta z_i = 0$	$\Delta y_i \oplus \sum_{j=1}^{n} P_{i,j} \cdot \Delta x_j = 0$	$\overline{S}(\Delta x_i, \Delta y_i) = 0$
Note	include *2n* linear equations	include *n* linear equations	include *n* linear equations	include *n* formal equations, only represent that Δx_i and Δy_i have some relations.

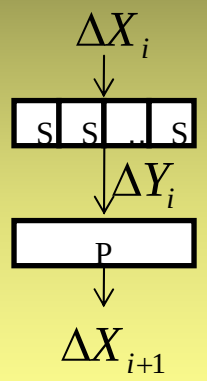
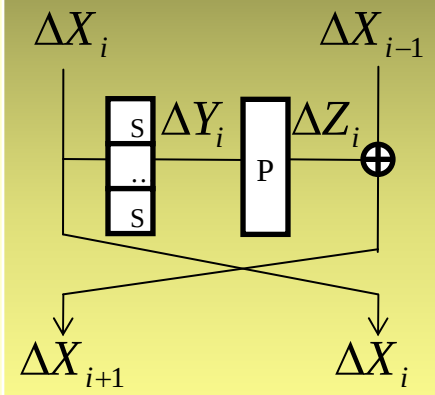
New Tool for Finding Impossible Differentials

——Difference propagation system

- **Difference propagation systems for word-oriented block ciphers**
  - Combine basic primitives together to build a system of equations
  - Choose *r*-round SPN ciphers and Feistel ciphers with SPN round functions as examples

New Tool for Finding Impossible Differentials

——Difference propagation system

Block ciphers	Difference propagation system	Note
	$\begin{cases} \overline{S}(\Delta X_{i,j}, \Delta Y_{i,j}) = 0, \\ \Delta X_{i+1}^T \oplus P \cdot \Delta Y_i^T = 0 \end{cases}$ $(1 \le i \le r, 1 \le j \le n)$	$N = (2r+1)n$ variables rn linear equations rn formal equations from Sbox layers
	$\begin{cases} \overline{S}(\Delta X_{i,j}, \Delta Y_{i,j}) = 0, \\ \Delta Z_i^T \oplus P \cdot \Delta Y_i^T = 0, \\ \Delta X_{i-1} \oplus \Delta Z_i \oplus \Delta X_{i+1} = 0 \end{cases}$ $(1 \le i \le r, 1 \le j \le n)$	$N = (3r+2)n$ variables $2rn$ linear equations rn formal equations from Sbox layers

New Tool for Finding Impossible Differentials

——Predict information and detect contradictions

- **Predict information from a difference propagation system**
  - a difference propagation system can be divided into two subsystems — $\mathcal{L}$ and $\mathcal{NL}$
    - ✓ $\mathcal{L}$ includes all linear equations
    - ✓ $\mathcal{NL}$ includes all formal equations from the Sbox layers

Example 1. For an *r*-round SPN cipher, we have

$$\begin{cases} \overline{S}(\Delta X_{i,j}, \Delta Y_{i,j}) = 0 \ (1 \le i \le r, 1 \le j \le n), \\ \Delta X_{i+1}^{T} \oplus P \cdot \Delta Y_{i}^{T} = 0 \ (1 \le i \le r, 1 \le j \le n) \end{cases}$$

→ System $\mathcal{NL}$

→ System $\mathcal{L}$

New Tool for Finding Impossible Differentials

——Predict information and detect contradictions

❑ Predict information from $\mathcal{NL}$

Lemma 1. Suppose S is a bijective Sbox, x is its input difference and y is its output difference, then

$x=0$ ($\neq 0$) if and only if $y=0$ ($\neq 0$).

❑ Predict information from $\mathcal{L}$

- If linear system $\mathcal{L}$ has solutions, we can solve it by Gauss-Elimination algorithm and recover information from the remaining system

Lemma 2. After solving a linear system by Gauss-Elimination algorithm, then
1) If an affine equation with only a variable, i.e., $x \oplus c=0$ (c is a constant), is found in the system, then $x=0$ if $c=0$ and $x\neq 0$ if $c\neq 0$;
2) If a linear equation with two variables, i.e., $x \oplus y=0$, is found in the system, then $x\neq 0$ if and only if $y\neq 0$.

New Tool for Finding Impossible Differentials

——Predict information and detect contradictions

Example 2. Suppose we have known that $u=0, x\neq 0$, and the difference propagation system is

$$\begin{cases} \overline{S}(u,v)=0, \\ x\oplus y=0, \\ x\oplus y\oplus z=0. \end{cases}$$

After one step of information prediction, we have

$$\begin{cases} \overline{S}(u,v)=0, \\ x\oplus y=0, \\ x\oplus y\oplus z=0. \end{cases} \xrightarrow{\text{Lemma 1}} \boxed{v=0} \qquad \begin{cases} x\oplus y=0, \\ x\oplus y\oplus z=0. \end{cases} \xrightarrow{\text{Gauss-Elimination}} \begin{cases} x\oplus y=0, \\ z=0. \end{cases} \xrightarrow{\text{Lemma 2}} \boxed{\begin{array}{l} y\neq 0, \\ z=0 \end{array}}$$

New Tool and Experimental Results

——Predict information and detect contradictions

❑ Detect contradictions

Proposition 1. For given plaintext difference ΔP and ciphertext difference ΔC, $\Delta P \to \Delta C$ is impossible if one of the following two situations happens:

1) The linear system $\mathcal{L}$ doesn't have any solution. That is, the rank of its coefficient matrix is not equal to the rank of its augmented matrix;

2) There exists a variable with both zero and nonzero values.

New Tool and Experimental Results

——Predict information and detect contradictions

- **A tiny example of the second type of contradiction**

Example 3. Suppose we have known that $x=0, z\neq 0$, and the following equations are included in a difference propagation system

$$\begin{cases} \overline{S}(x,y)=0, \\ y\oplus z=0. \end{cases}$$

After a step of information prediction, we have

$$\left.\begin{cases} \overline{S}(x,y)=0, \xrightarrow{\text{Lemma 1}} y=0 \\ y\oplus z=0. \xrightarrow{\text{Lemma 2}} y\neq 0 \end{cases}\right\} \text{Contradiction!}$$

New Tool for Finding Impossible Differentials

——Algorithm complexity and experimental results

❑ Complexity

1. Build the difference propagation system of a block cipher;
2. **for** *each pair of* $(\Delta P, \Delta C)$ *we choose* **do**;
3. *index*:=true;
4. **while** *index* **do**;
5. Predict information from the difference propagation system;
6. **if** *a contradiction is found* **then**;
7. **return** true; **break**;
8. **elseif** *cannot get any new information* **then**;
9. *index*:=false;
10. **end if**;
11. **end while**;
12. **end for**;

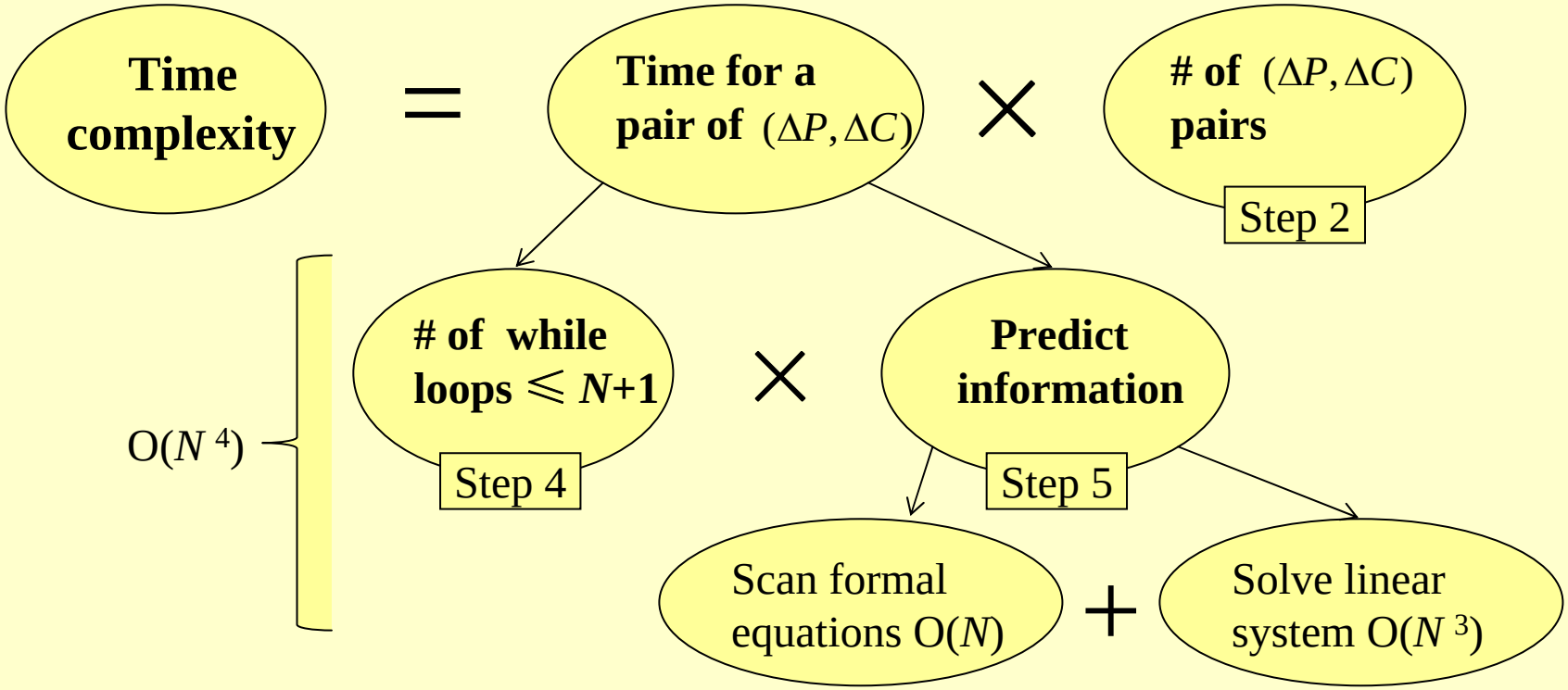
Memory complexity: store the difference propagation system

New Tool for Finding Impossible Differentials

——Algorithm complexity and experimental results

2. **for** *each pair of* $(\Delta P, \Delta C)$ *we choose* **do**;
4. **while** *index* **do**;
5. Predict information from the difference propagation system;

Suppose N is the number of variables involved in a difference propagation system.



New Tool for Finding Impossible Differentials

——Algorithm complexity and experimental results

Experimental results

- Implemented our tool on a 2.66 GHz processor with MAGMA package, and applied it to many word-oriented block ciphers.
- It costs about 2-3 hours to enumerate 2^{16} pairs of $(\Delta P, \Delta C)$.
- Results obtained by our tool have been shown in a previous table (see our contributions).
- More detail results, together with the source code of our tool for SPN ciphers and Feistel ciphers, have been given in the full version, which is available at eprint.iacr.org/2012/214.

New Tool for Finding Impossible Differentials

——Discussions

Correctness

- Impossible differentials found by our tool must be correct, since we use sufficient conditions to detect them.

Limitations

- Our tool is not able to exploit any properties of the Sboxes beyond the fact that they are bijective.
- It may fail if a block cipher is not word-oriented or uses an Sbox that is not bijective.
- Note: these limitations also exist in the U-method and the UID-method.

Outline

- Introduction
- New Tool for Finding Impossible differentials
- Conclusions

Conclusions

——Conclusions and future work

Conclusions

- A new tool for finding impossible differentials is proposed.
- Although our tool does not improve the length of impossible differentials for existing block ciphers,
  - ✓ It helps in reducing the gap between previous tools and ad hoc approaches
  - ✓ It finds many new impossible differentials, which may improve known attacks.

Future work

- Find more general filtering conditions to detect contradictions
- Exploit any properties of the Sboxes beyond the fact that they are bijective.

Thank you!